

DDDDDDDD	UU	UU	PPPPPPPP	11	11	
DDDDDDDD	UU	UU	PPPPPPPP	11	11	
DD	DD	UU	UU	PP	PP	1111
DD	DD	UU	UU	PP	PP	1111
DD	DD	UU	UU	PP	PP	11
DD	DD	UU	UU	PP	PP	11
DD	DD	UU	UU	PPPPPPPP		11
DD	DD	UU	UU	PPPPPPPP		11
DD	DD	UU	UU	PP		11
DD	DD	UU	UU	PP		11
DD	DD	UU	UU	PP		11
DD	DD	UU	UU	PP		11
DD	DD	UU	UU	PP		11
DDDDDDDD	UUUUUUUUUU	UUUUUUUUUU	PP	111111	111111	111111
DDDDDDDD	UUUUUUUUUU	UUUUUUUUUU	PP	111111	111111	111111

```

LL          IIIII
LL          IIIII
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LLLLLLLLLLL IIIII
LLLLLLLLLLL IIIII

SSSSSSSSS
SSSSSSSSS
SS
SS
SS
SS
SSSSSS
SSSSSS
SS
SS
SS
SS
SSSSSSSSS
SSSSSSSSS

```

DUF

027
027
027
027
027
027
027
028
028
028
028
028
028
028
028
029
029
029
029
029
029
029
029
029
030
030
030
030
030
030
030
030
030
030
031
031
031
031
031
031
031
031
031
031
032
032
032
032
032
032
032
032

B 15
16-Sep-1984 00:21:20
5-Sep-1984 13:53:59

VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]DUP11.FOR;1

Page 1

```
0001      SUBROUTINE DUP11 (lun,parcsr,rxcsr,rxdbuf,txcsr,txdbuf)
0002      C
0003      C Version:      'V04-000'
0004      C
0005      C*****
0006      C*
0007      C*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0008      C*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0009      C*  ALL RIGHTS RESERVED.
0010      C*
0011      C*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0012      C*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0013      C*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0014      C*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0015      C*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0016      C*  TRANSFERRED.
0017      C*
0018      C*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0019      C*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0020      C*  CORPORATION.
0021      C*
0022      C*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0023      C*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0024      C*
0025      C*
0026      C*****
0027      C
0028      C
0029      C
0030      C      Author: Sharon Reynolds      Creation date: 8-Sept-80
0031      C
0032      C
0033      C
0034      C      Functional description:
0035      C
0036      C      This module decodes the Dup11 register contents. It is called
0037      C      by the DUP3271 module.
0038      C
0039      C      Modified by:
0040      C
0041      C      V03-002 SAR0112      Sharon A. Reynolds,      23-Jun-1983
0042      C      Changed the carriage control in the 'format' statements
0043      C      for use with ERF.
0044      C
0045      C      v03-001 BP0001      Brian Porter,      20-AUG-1982
0046      C      Minor edit.
0047      C**
0048      C--
0049      C
0050      C
0051      C
0052      C
0053      C      Byte      lun
0054      C
0055      C      Integer*4      addrs
0056      C      Integer*4      char
0057      C      Integer*4      field
```



```
0058 Integer*4 temp
0059 Integer*4 rxcsr
0060 Integer*4 rxdbuf
0061 Integer*4 txcsr
0062 Integer*4 txdbuf
0063 Integer*4 parcsr
0064 Integer*4 compressC
0065 Integer*4 compress4
0066
0067 logical*1 diagnostic_mode
0068
0069 Character*23 protocol(0:1)
0070 Character*12 oper_mode(0:1)
0071 Character*18 rxcsr_1(0:0)
0072 Character*26 rxcsr_2(3:8)
0073 Character*24 rxcsr_3(10:11)
0074 Character*18 rxcsr_4(14:15)
0075 Character*26 rxdbuf_1(8:10)
0076 Character*19 rxdbuf_2(12:12)
0077 Character*17 rxdbuf_3(14:15)
0078 Character*5 txcsr_1(4:4)
0079 Character*29 txcsr_2(6:9)
0080 Character*28 txcsr_3(15:15)
0081 Character*27 txdbuf_1(8:10)
0082 Character*12 parcsr_1(9:9)
0083 Character*26 dtr_status(0:1)
0084 Character*22 rts_status(0:1)
0085 Character*21 dsr_status(0:1)
0086 Character*21 car_status(0:1)
0087 Character*20 cts_status(0:1)
0088
0089
0090
```

```
0091 C
0092 C Define text for bits in the RXCSR register
0093 C
```

```
0094
0095 Data rxcsr_1(0) /*DATA SET CHANGE B*/
0096 Data rxcsr_2(3) /*SECONDARY TRANSMIT DATA*/
0097 Data rxcsr_2(4) /*RECEIVER ENABLE*/
0098 Data rxcsr_2(5) /*DATA SET INTERRUPT ENABLE*/
0099 Data rxcsr_2(6) /*RECEIVER INTERRUPT ENABLE*/
0100 Data rxcsr_2(7) /*RECEIVER DONE*/
0101 Data rxcsr_2(8) /*STRIP SYNC CHARACTER*/
0102 Data rxcsr_3(10) /*SECONDARY RECEIVED DATA*/
0103 Data rxcsr_3(11) /*RECEIVER ACTIVE*/
0104 Data rxcsr_4(14) /*RING*/
0105 Data rxcsr_4(15) /*DATA SET CHANGE A*/
0106
0107
```

```
0108 C
0109 C Define bits for MODEM/LINE STATUS from rxcsr register
0110 C
```

```
0111
0112 Data dtr_status(0) /*DATA TERMINAL READY = OFF*/
0113 Data dtr_status(1) /*DATA TERMINAL READY = ON*/
0114
```



```
0115      Data   rts_status(0)  /*REQUEST TO SEND = OFF*/
0116      Data   rts_status(1)  /*REQUEST TO SEND = ON*/
0117
0118      Data   dsr_status(0)    /*DATA SET READY = OFF*/
0119      Data   dsr_status(1)    /*DATA SET READY = ON*/
0120
0121      Data   car_status(0)    /*CARRIER DETECT = OFF*/
0122      Data   car_status(1)    /*CARRIER DETECT = ON*/
0123
0124      Data   cts_status(0)    /*CLEAR TO SEND = OFF*/
0125      Data   cts_status(1)    /*CLEAR TO SEND = ON*/
0126
0127
0128      C
0129      C      Define text for bits in the RXDBUF register
0130      C
0131
0132      Data   rxdbuf_1(8)      /*START OF RECEIVED MESSAGE*/
0133      Data   rxdbuf_1(9)      /*END OF RECEIVED MESSAGE*/
0134      Data   rxdbuf_1(10)     /*RECEIVER ABORT*/
0135      Data   rxdbuf_2(12)     /*RECEIVER CRC ERROR*/
0136      Data   rxdbuf_3(14)     /*RECEIVER OVERRUN*/
0137      Data   rxdbuf_3(15)     /*RECEIVER ERROR*/
0138
0139
0140      C
0141      C      Define text for bits in the TXCSR register
0142      C
0143
0144      Data   oper_mode(0)     /*FULL DUPLEX*/
0145      Data   oper_mode(1)     /*HALF DUPLEX*/
0146      Data   txcsr_1(4)       /*SEND*/
0147      Data   txcsr_2(6)       /*TRANSMITTER INTERRUPT ENABLE*/
0148      Data   txcsr_2(7)       /*TRANSMITTER DONE*/
0149      Data   txcsr_2(8)       /*DEVICE RESET*/
0150      Data   txcsr_2(9)       /*TRANSMITTER ACTIVE*/
0151      Data   txcsr_3(15)      /*TRANSMITTER DATA LATE ERROR*/
0152
0153
0154      C
0155      C      Define text for bits in the TXDBUF register
0156      C
0157
0158      Data   txdbuf_1(8)      /*TRANSMIT START OF MESSAGE*/
0159      Data   txdbuf_1(9)      /*END OF TRANSMITTED MESSAGE*/
0160      Data   txdbuf_1(10)     /*TRANSMIT ABORT*/
0161
0162
0163      C
0164      C      Define text for bits in the PARCSR register
0165      C
0166
0167      Data   parcsr_1(9)       /*CRC INHIBIT*/
0168      Data   protocol(0)       /*BIT ORIENTED PROTOCOL*/
0169      Data   protocol(1)       /*BYTE ORIENTED PROTOCOL*/
0170
0171
```

```
0172
0173
0174
0175      diagnostic_mode = .false.
0176
0177      if (lib$extzv(11,2,txcsr) .ne. 0) diagnostic_mode = .true.
0178
0179      C
0180      C
0181      C
0182      Call LINCHK (lun,2)
0183      Write (lun,20) rxcsr
0184      Format (' ',T8,'RXCSR',T24,Z8.4)
0185
0186      if (.not. diagnostic_mode) then
0187
0188      Call LINCHK (lun,1)
0189      Write (lun,30) (' ',I=1,34)
0190      Format (' ',T40,Z4A1)
0191
0192      Call LINCHK (lun,1)
0193      Write (lun,40)
0194      Format (' ',T47,'MODEM/LINE STATUS')
0195
0196      Field=LIB$EXTZV (1,1,rxcsr)
0197
0198      Call LINCHK (lun,1)
0199      Write (lun,50) dtr_status(field)
0200      Format (' ',T40,A<compressC (dtr_status(field)))>>
0201
0202
0203      Field=LIB$EXTZV (2,1,rxcsr)
0204
0205      Call LINCHK (lun,1)
0206      Write (lun,60) rts_status(field)
0207      Format (' ',T40,A<compressC (rts_status(field)))>>
0208
0209
0210      Field=LIB$EXTZV (9,1,rxcsr)
0211
0212      Call LINCHK (lun,1)
0213      Write (lun,70) dsr_status(field)
0214      Format (' ',T40,A<compressC (dsr_status(field)))>>
0215
0216
0217      Field=LIB$EXTZV (12,1,rxcsr)
0218
0219      Call LINCHK (lun,1)
0220      Write (lun,80) car_status(field)
0221      Format (' ',T40,A<compressC (car_status(field)))>>
0222
0223
0224      Field=LIB$EXTZV (13,1,rxcsr)
0225
0226      Call LINCHK (lun,1)
0227      Write (lun,90) cts_status(field)
0228      Format (' ',T40,A<compressC (cts_status(field)))>>
```



```
0229
0230      Call LINCHK (lun,1)
0231      Write (lun,100) ('*',I=1,34)
0232 100      Format (' ',T40,34A1)
0233
0234      Call OUTPUT (lun,rxcsr,rxcsr_1,0,0,0,'0')
0235
0236      Call OUTPUT (lun,rxcsr,rxcsr_2,3,3,8,'0')
0237
0238      Call OUTPUT (lun,rxcsr,rxcsr_3,10,10,11,'0')
0239
0240      Call OUTPUT (lun,rxcsr,rxcsr_4,14,14,15,'0')
0241      endif
0242
0243      C
0244      C
0245      C
0246
0247      Call LINCHK (lun,1)
0248      Write (lun,110) rxdbuf
0249 110      Format (' ',T8,'RXDBUF',T24,Z8.4)
0250
0251      if (.not. diagnostic_mode) then
0252
0253      Field=LIB$EXTZV(0,8,rxdbuf)
0254
0255      Call LINCHK (lun,1)
0256      Write (lun,120) field
0257 120      Format (' ',T40,'RECEIVER DATA BUFFER = ',
0258      1 I<compress4 (field)>,>,'.')
0259
0260      Call OUTPUT (lun,rxdbuf,rxdbuf_1,8,8,10,'0')
0261
0262      Call OUTPUT (lun,rxdbuf,rxdbuf_2,12,12,12,'0')
0263
0264      Call OUTPUT (lun,rxdbuf,rxdbuf_3,14,14,15,'0')
0265      endif
0266
0267      C
0268      C
0269      C
0270
0271      Call LINCHK (lun,1)
0272      Write (lun,130) txcsr
0273 130      Format (' ',T8,'TXCSR',T24,Z8.4)
0274
0275      if (.not. diagnostic_mode) then
0276
0277      Field=LIB$EXTZV(3,1,txcsr)
0278
0279      Call LINCHK (lun,1)
0280      Write (lun,140) oper_mode(field)
0281 140      Format (' ',T40,'OPERATION MODE = ',
0282      1 A<compressC (oper_mode(field))>>')
0283
0284      Call OUTPUT (lun,txcsr,txcsr_1,4,4,4,'0')
0285
```

```
0286      Call OUTPUT (lun,txcsr,txcsr_2,6,6,9,'0')
0287
0288      Call OUTPUT (lun,txcsr,txcsr_3,15,15,15,'0')
0289      else
0290
0291      Call LINCHK (lun,1)
0292      Write (lun,150)
0293 150      Format (' ',T40,'DIAGNOSTIC MODE')
0294      endif
0295
0296      C
0297      C      Decode transmitter data buffer register
0298      C
0299
0300      Call LINCHK (lun,1)
0301      Write (lun,160) txdbuf
0302 160      Format (' ',T8,'TXDBUF',T24,Z8.4)
0303
0304      if (.not. diagnostic_mode) then
0305
0306      Field=LIBSEXTZV(0,8,txdbuf)
0307
0308      Call LINCHK (lun,1)
0309      Write (lun,170) field
0310 170      Format (' ',T40,'TRANSMITTER DATA BUFFER = ',
0311      1 I<compress4 (field)>,'.')
0312
0313      Call OUTPUT (lun,txdbuf,txdbuf_1,8,8,10,'0')
0314      endif
0315
0316      C
0317      C      Decode and output PARCSR register contents
0318      C
0319
0320
0321
0322      Call LINCHK (lun,1)
0323      Write (lun,180) parcsr
0324 180      Format (' ',T8,'PARCSR',T24,Z8.4,/,
0325      1 T40,'** LOADED WITH **')
0326
0327      if (.not. diagnostic_mode) then
0328
0329      Call OUTPUT (lun,parcsr,parcsr_1,9,9,9,'0')
0330
0331      Temp=LIBSEXTZV (12,1,parcsr)
0332
0333      Field=LIBSEXTZV (15,1,parcsr)
0334
0335      Call LINCHK (lun,1)
0336      Write (lun,190) protocol(field)
0337 190      Format (' ',T40, A<compressC (protocol(field))> )
0338
0339      Call LINCHK (lun,1)
0340      If (field .eq. 0 .and. temp .eq.0) then
0341
0342      Write (lun,200)
```


DUP11

H 15
16-Sep-1984 00:21:20
5-Sep-1984 13:53:59

VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]DUP11.FOR;1

Page 7

```
0343 200  Format (' ',T40,'OPERATING AS A PRIMARY STATION')
0344
0345      Else if (field .eq. 0 .and. temp .eq. 1) then
0346
0347      Write (lun,210)
0348 210  Format (' ',T40,'OPERATING AS A SECONDARY STATION')
0349
0350      Addr=LIB$EXTZV (0,8,parcsr)
0351
0352      Call LINCHK (lun,1)
0353      Write (lun,220) addr
0354 220  Format (' ',T40,'SECONDARY STATION ADDRESS = ',
0355          1 I<Compress4 (addr)>,'.')
0356
0357      Else
0358
0359      Char=LIB$EXTZV (0,8,parcsr)
0360
0361      Write (lun,230) char
0362 230  Format (' ',T40,'EXPECTED SYNC CHAR = ',
0363          1 I<Compress4 (char)>,'.')
0364
0365
0366      Endif
0367      endif
0368
0369      Return
0370
0371
0372      END
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	2023	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	592	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	2024	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
Total Space Allocated	4639	

ENTRY POINTS

Address	Type	Name
0-00000000		DUP11

VARIABLES

Address	Type	Name	Address	Type	Name
2-0000039C	I*4	ADDRS	2-000003A0	I*4	CHAR
2-00000399	L*1	DIAGNOSTIC_MODE	2-000003A4	I*4	FIELD
2-000003AC	I*4	I	AP-00000004a	L*1	LUN
AP-00000008a	I*4	PARCSR	AP-0000000Ca	I*4	RXCSR
AP-00000010a	I*4	RXDBUF	2-000003A8	I*4	TEMP
AP-00000014a	I*4	TXCSR	AP-00000018a	I*4	TXDBUF

ARRAYS

Address	Type	Name	Bytes	Dimensions
2-00000347	CHAR	CAR_STATUS	42	(0:1)
2-00000371	CHAR	CTS_STATUS	40	(0:1)
2-0000031D	CHAR	DSR_STATUS	42	(0:1)
2-000002BD	CHAR	DTR_STATUS	52	(0:1)
2-0000002E	CHAR	OPER_MODE	24	(0:1)
2-000002B1	CHAR	PARCSR_1	12	(9:9)
2-00000000	CHAR	PROTOCOL	46	(0:1)
2-000002F1	CHAR	RTS_STATUS	44	(0:1)
2-00000046	CHAR	RXCSR_1	18	(0:0)
2-00000058	CHAR	RXCSR_2	156	(3:8)
2-000000F4	CHAR	RXCSR_3	48	(10:11)
2-00000124	CHAR	RXCSR_4	36	(14:15)
2-00000148	CHAR	RXDBUF_1	78	(8:10)
2-00000196	CHAR	RXDBUF_2	19	(12:12)
2-000001A9	CHAR	RXDBUF_3	34	(14:15)
2-000001CB	CHAR	TXCSR_1	5	(4:4)
2-000001D0	CHAR	TXCSR_2	116	(6:9)
2-00000244	CHAR	TXCSR_3	28	(15:15)
2-00000260	CHAR	TXDBUF_1	81	(8:10)

DUP11

J 15
16-Sep-1984 00:21:20
5-Sep-1984 13:53:59

VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]DUP11.FOR;1

Page 9

LABELS

Address	Label	Address	Label	Address	Label	Address	Label	Address	Label	Address	Label
1-0000003B	20'	1-0000004E	30'	1-00000058	40'	1-00000071	50'	1-0000007D	60'	1-00000089	70'
1-00000095	80'	1-000000A1	90'	1-000000AD	100'	1-000000B7	110'	1-000000CA	120'	1-000000F2	130'
1-00000104	140'	1-00000123	150'	1-0000013A	160'	1-0000014D	170'	1-00000178	180'	1-000001A3	190'
1-000001AF	200'	1-000001D5	210'	1-000001FD	220'	1-0000022A	230'				

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name	Type	Name	Type	Name
I*4	COMPRESS4	I*4	COMPRESSC	I*4	LIB\$EXTZV		LINCHK		OUTPUT

COMMAND QUALIFIERS

FORTRAN /LIS=LIS\$:DUP11/OBJ=OBJ\$:DUP11 MSRC\$:DUP11

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)

/DEBUG=(NOSYMBOLS,TRACEBACK)

/STANDARD=(NOSYNTAX,NOSOURCE_FORM)

/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)

/F77 /NOG_FLOATING /I4 /OPTIMIZE /WARNINGS /NOD_LINES /NOCROSS_REFERENCE /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time:	5.30 seconds
Elapsed Time:	13.85 seconds
Page Faults:	201
Dynamic Memory:	200 pages

0147 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

